Interactive Stroke-Based NPR using Hand Postures on Large Displays

Grubert, Jens; Carpendale, Sheelagh; Isenberg, Tobias

Published in:
EPRINTS-BOOK-TITLE

Publication date:
2008

Interactive Stroke-Based NPR using Hand Postures on Large Displays

Jens Grubert[1] Sheelagh Carpendale[1] Tobias Isenberg[1,2]

[1]University of Calgary, Canada [2]University of Groningen, The Netherlands

Abstract

We explore the use of hand postures to interact with stroke-based rendering (SBR) on touch-sensitive large displays. In contrast to traditional WIMP interfaces, we allow people to directly engage with and influence a rendering. Our system allows the creation of new stroke primitives as well as provides mechanisms to distribute and then manipulate them on the canvas. We offer a set of natural mappings from hand postures to rendering parameterizations. The resulting system allows an intuitive exploration of SBR without the need for traditional desktop interfaces.

Categories and Subject Descriptors (according to ACM CCS): H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction styles, Input devices and strategies; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques; I.3.m [Computer Graphics]: Miscellaneous—Non-photorealistic rendering.

1. Introduction and Motivation

Work in non-photorealisitic rendering (NPR) has produced a wide variety of visual styles [GG01, SS02] and great results have been achieved through further refinement and adjustment of these methods. However, the style of a rendering is still usually adjusted through in-direct manipulation of parameters—using techniques such as sliders, checkboxes, buttons, etc. Working toward a more exploratory and less cumbersome approach to adjusting rendering styles and parameters and to inspire creativity, we explore the interaction with and parameterization of a NPR system using a large, touch-sensitive display. Our interface is based on combining pen use with a small set of hand postures (see Fig. 1).

We extend the Interactive Canvas [SIMC07] that supports the creation of stroke-based renderings (SBR, [Her03]). The previous system only allowed indirect interaction through a set of widgets: a palette, tools, and a pre-defined set of primitives. During informal observations of the system's use, we discovered that this type of interaction was not sufficient to allow people to express their creative ideas. Several people expressed, in particular, the desire to create their own strokes in order to have more creative freedom. One person mentioned, e. g., that it is "hard to find a good cloud texture." When more formally studying [GHC*07] an adapted system with an on-screen menu controlled by a separate Wii Nunchuck instead of the palette we gathered that participants

Figure 1: *Hand postures in use and a pen as physical prop.*

had problems with recalling even a few actions (out of eight in this case; "you need a good memory to recall all the options"). A positive aspect of the adapted system was that participants felt immersed in their images through directly touching the screen without having to look for menus or a keyboard to control their interactions ("feels like a physical connection"). Even though only a single hand posture was used at this point to interact with the screen, participants said it "feels like finger painting." These finding encouraged us to explore a limited set of direct hand interactions.

2. Related Work

Both the Interactive Canvas [SIMC07] and our hand posture approach are based on representing properties of the interface in interaction buffers [IMC06]. These buffers are used to facilitate the interactive manipulation of properties for a large number of primitives. In contrast to the Interactive Canvas, however, we employ only *instantaneous buffers* to communicate changes of attributes rather than representing the attributes directly. The only exception to this rule is the color buffer which can store the actual images that are used to derive the color values for the strokes at their local position.

In addition, there are a number of other approaches that have influenced our work. In particular, work that explores the interaction with NPR has inspired us such as the WYSIWYG-NPR system [KMM*02]. WYSIWYG-NPR supports interactive re-stylization of strokes automatically extracted from 3D scenes as well as strokes hand-drawn onto the surface of objects. Lang et al. [LFS03] use digitally enhanced real brushes to allow users to interact with a digital painting system. Ryokai et al.'s I/O Brush system [RMI04] goes one step further and embeds a camera into a physical brush to allow users to capture real-world images which then can be used for painting in a digital system. Finally, Lee et al. [LOG06] introduced fluid jet painting and showed how it can be used interactively on a digital tabletop display to create images in the style of Jackson Pollock.

Vision-based hand posture and hand gesture detection has received considerable attention in recent years. As many techniques were specifically designed for 3D tracking, only few approaches are applicable for direct touch interfaces. The one most relevant for our work is [vHB01]. Here, finger tracking and hand posture detection are used for interacting via bare hands on a front-projected direct touch display. Hand postures are used, e. g., for supporting presentations or brainstorming. Even though the total number of recognizable hand postures is larger than in our approach, there are some restrictions. In particular, the hand posture detection and finger tracking relies on front-projected displays as additional hardware (IR LEDs, camera) is required behind the display, its accuracy depends on the projected image, and requires additional image processing. Our approach, in contrast, is independent from the display technology and the displayed image, uses only commercially available touch technology, and does not require additional image processing.

3. Stroke-Based Rendering with Hand Postures

While stroke-based rendering [Her03] thus far has often been used for the automatic generation of NPR images, it lends itself naturally to exploratory interaction approaches as it easily translates from traditional drawing and painting. This possibility for interaction invites the use of direct-manipulation techniques using touch-sensitive walls or tables. We focus on touch interaction to support such direct interaction possibilities including, e. g., finger painting. Typical direct-touch interaction uses one finger to simulate a mouse-click (touch-down). By employing a set of hand postures, however, we explore a much larger spectrum of direct hand interaction techniques. The postures allow us to both sketch actual strokes as well as to distribute these on the canvas and modify their properties in a finger painting way. This way we make use of the hand's capabilities, going beyond a mere pointing for invoking interactions.

3.1. Motivating and Designing Hand Postures

One motivation for employing hand postures for interacting with SBR arose from how artists use finger painting. In regular painting a brush is used to transfer paint onto a canvas. A finger painter, instead, uses single fingers, multiple fingers or even the whole hand to apply paint and to modify it (e. g., smearing it) once it is on the canvas and achieves a lot of artistic freedom with this. Another reason for using hand postures instead of on-screen menus to enable interaction with SBR is the reduction of visual clutter and the possibility to provide a minimalistic interface. People are easily able to remember a small number of hand postures and we can, thus, trade in visible interface elements for a closer and more immediate 'invisible' interaction with the digital canvas.

The design of hand postures for interaction was guided by artists who work in or regularly visit our lab, by our goal to provide intuitive interaction that is easily learned, and by memory and technical constraints. Our technical setup comprises a SMART Technology DViT that provides touch sensitivity for large projected, plasma, or LC displays. This technology uses four cameras in the display's corners which see the shadow of an interacting object (e. g., hand or pen) in front of a strip of infrared LEDs along the side of the display. This yields the center position as well as the approximated width and height of the object which we use in turn for recognizing hand positions and postures.

Inaccuracies during the recognition process of interaction positions and their parameters limit the number and type of possible hand postures. For example, the physical area that is covered by one hand posture has to be disjoint from areas covered by other postures. In addition, we accounted for individual differences between people's hand and finger sizes for our design. Finally, we also considered what postures people are able to form easily with their hands and which ones are used for natural interactions with other people. Examples for such natural postures include the use of one finger for pointing or forming a fist for showing emphasis.

Based on these constraints, we developed four hand postures that can be discriminated by our hardware setup for a variety of hand sizes and that can be formed easily with a single hand: pointing with one finger, pointing with two fingers, a fist, and a flat hand (shown in Fig. 2).

While this small number of distinguishable natural hand postures is advantageous in that they are easy to remember, they do limit the number of functionalities that we could map while still avoiding menus or keyboard interactions. Also, the postures did not lend themselves intuitively to encode the distinction between sketching new stroke primitives and using these primitives by applying them to the canvas. In general, we envisioned the posture-to-functionality mapping to make sense to people in the context of directly-manipulative stroke-based rendering to avoid a lengthy learning process.

Therefore, we made use of a pen tray that is attached to SMART Technology's DViT boards (see Fig. 1, right) which can detect when a pen is lifted from its associated pocket and when it is placed back. While technically it would be possible to distinguish four different pens and, thus, map four more sets of functionalities to the four hand postures, we felt that this might make it hard to remember the mappings. Thus, we used only one additional set of mappings, indicated by any pen being lifted from its pocket.

This action was consequently mapped to naturally indicate the shift from distributing stroke primitives on the canvas (pen down) to sketching a new stroke primitive (pen lifted) because the use of a pen can easily be associated with a sketching action. In addition, the fact that one pen is lifted during the sketching actions also enabled us to add an additional "hand posture," i. e. the pen, and use it for even finer control (tip diameter ca. 3 mm) than a single pointing finger.

3.2. Interacting through Hand Postures

Associating postures with an increasingly small contact area with actions that have increasing precision was a general design decision for developing a natural mapping of hand postures to interactions. This let people employ postures that do not obstruct much of the display for fine actions; use postures that cover more space to initiate stronger actions; and interact more quickly with more primitives that need less accuracy. In addition, whether creating new strokes (pen lifted) or applying these strokes (pen down), the hand postures are mapped to actions that are as similar as possible.

In the stroke sketching mode, the pen creates the finest line, followed by one pointing finger and two pointing fingers with increasingly stronger strokes, while the strongest stroke is created by employing a fist. Fig. 3 shows the results of using these four different postures for sketching strokes (left; pen to fist) and how they appear once they are distributed (Fig. 3, right). Similar to this set of mappings, in the stroke distribution mode the one finger posture is mapped to creating few strokes in a small radius around the finger and a fist creates many strokes in a larger radius.

In distribution mode, however, the two finger posture is

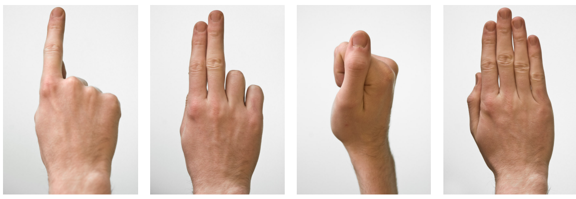

Figure 2: *Hand postures: 1 finger, 2 fingers, fist, and hand.*

Figure 3: *Strokes can be sketched with pen, one finger, two fingers, fist (left side). Strokes are rendered to texture and distributed with the one finger posture (shown on the right side) or the fist posture (this would produce more strokes within a larger region). The different colors of the strokes on the left side are derived from the underlying color image.*

Figure 4: *A sketched stroke (left) is first distributed with one finger, aligned with two fingers, faded with the hand, and redistributed with the fist posture (right, from top to bottom).*

mapped to stroke alignment which can be used to organize strokes once they have been applied, for instance to emphasize certain structures in the image. While automatic alignments could certainly be derived, e. g., from image edges or gradients [Her03], our interactive alignment allows exploration of stroke alignment, providing more flexibly and creative freedom. Predominantly linear features can easily be created by sketching a stroke in alignment direction, distributing it on the canvas, and finally using the alignment posture. Here the two fingers of the posture serve as a natural indication of the alignment direction.

The mapping of the full hand posture is inspired by the erasing commonly used on black and white boards and deletes strokes or parts of strokes from the sketch. Similarly for distributing primitives, the full hand posture is used to erasing strokes, albeit it first makes them more transparent before deleting them entirely. This addresses comments we received in our previous study [GHC*07] in which participants expressed the desire for "subtle erasing." The fading out can be interrupted at any time by moving the hand away from the affected strokes, making it possible to change stroke appearance from opaque to transparent. Fig. 4 shows an example how a stroke is sketched, distributed, aligned, and faded out by means of the different postures.

3.3. Technical Realization

For detecting hand postures we solely rely on the available functionality of the Smartboard DViT, avoiding the overhead of additional image processing. We use the screen position and the approximated values for width and height of the bounding box of an interacting object. With these parameters we compute the touched area and use that to determine the hand posture. While processing area values we have to accommodate inaccuracies and jitter during the initial acquisition stage since stability of incoming values cannot be guaranteed. We compensate for this effect by averaging the first four width and height values we receive as input. Furthermore, we have to account for the fact that the same object has varying area values across the screen. The closer a touch happens to an edge or a corner of the screen the smaller are the delivered area values. Thus, we use the position in addition to width and height to determine a hand posture.

In order to provide a mechanism for drawing strokes that would work with just locations and areas as input, we implemented an adapted Dynadraw algorithm [Hae89]. In Dynadraw, one creates smooth, calligraphic strokes with a brush that follows a damped spring model and several brush parameters including mass, velocity, and friction influence the behaviour of the brush and, thus, the appearance of the resulting strokes. For these parameters we empirically derived settings for the different hand postures and specified a minimal and maximal stroke width to reflect the different characteristics. In addition, to avoid the abrupt ending of a stroke when the hand or the pen is lifted (see Fig. 3), we modified the behaviour of the resulting strokes at the beginning and at the end of a brush movement. The strokes are changed to end in a cone shape rather then in a broad, flat shape. The system was developed in C++ with OpenGL as rendering API.

4. Conclusion

Fig. 5 shows an example in which the technique was applied to a create an NPR version of a photograph. Notice how the different regions of the picture are expressed using a number of unique, self-created stroke primitives. While this technique may be useful for artists, we see its main user group in the general public: People will be able to experiment with their personal photographs as touch-sensitive displays are becoming increasingly common. The use of easy and intuitive hand postures will allow users to employ a range of styles without having to resort to looking up menu options.

Figure 5: *Using a background image (top) a stroke-based rendered image was created in less than 5 minutes (bottom).*

In summary, we have explored the use of natural hand postures to interact with and parameterize non-photorealistic stroke-based rendering. In addition, we provide the option to create one's own stroke primitives to further enhance the available creative freedom. We also suggest that the combination of posture-based interaction with freely definable stroke primitives gives our approach the character of creating collages with a wide variety of materials. In a way this pushes the boundaries of SBR—one could argue that collages are stroke-based rendering with larger primitives.

Acknowledgements

We thank our funding providors Alberta Igenuity, NSERC, CFI, iCORE, and SMART Technologies.

References

[GG01] Gooch B., Gooch A. A.: *Non-Photorealistic Rendering*. A K Peters, Ltd., Natick, 2001.

[GHC*07] Grubert J., Hanckock M., Carpendale S., Tse E., Isenberg T.: *Interacting with Stroke-Based Rendering on a Wall Display*. Tech. Rep. 2007-882-34, Dept. of Comp. Science, Univ. of Calgary, 2007.

[Hae89] Haeberli P.: Dynadraw. *Grafica OBSCURA* (July 1989). http://www.graficaobscura.com/dyna/.

[Her03] Hertzmann A.: A Survey of Stroke-Based Rendering. *IEEE CG&A 23*, 4 (July/Aug. 2003), 70–81.

[IMC06] Isenberg T., Miede A., Carpendale S.: A Buffer Framework for Supporting Responsive Interaction in Information Visualization Interfaces. In *Proc. C^5* (Los Alamitos, 2006), IEEE Computer Society, pp. 262–269.

[KMM*02] Kalnins R. D., Markosian L., Meier B. J., Kowalski M. A., Lee J. C., Davidson P. L., Webb M., Hughes J. F., Finkelstein A.: WYSIWYG NPR: Drawing Strokes Directly on 3D Models. *ACM ToG 21*, 3 (July 2002), 755–762.

[LFS03] Lang D., Findlater L., Shaver M.: CoolPaint: Direct Interaction Painting. In *UIST Posters* (2003).

[LOG06] Lee S., Olsen S. C., Gooch B.: Interactive 3D Fluid Jet Painting. In *Proc. NPAR* (New York, 2006), ACM Press, pp. 97–104.

[RMI04] Ryokai K., Marti S., Ishii H.: I/O Brush: Drawing with Everyday Objects as Ink. In *Proc. CHI* (New York, 2004), ACM Press, pp. 303–310.

[SIMC07] Schwarz M., Isenberg T., Mason K., Carpendale S.: Modeling with Rendering Primitives: An Interactive Non-Photorealistic Canvas. In *Proc. NPAR* (New York, 2007), ACM Press, pp. 15–22.

[SS02] Strothotte T., Schlechtweg S.: *Non-Photorealistic Computer Graphics. Modeling, Animation, and Rendering*. Morgan Kaufmann, San Francisco, 2002.

[vHB01] von Hardenberg C., Bérard F.: Bare-Hand Human-Computer Interaction. In *Proc. PUI* (New York, 2001), ACM Press. Posters & demos session.